INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Iterative Methods for Model Reduction by Domain Decomposition

Marcelo Buffoni — Haysam Telib — Angelo Iollo

N° 6383

Décembre 2007

Thème NUM

Rapport de recherche

N 0249-6399

Iterative Methods for Model Reduction by Domain Decomposition

Marcelo Buffoni[*] , Haysam Telib[†] , Angelo Iollo[*]

Thème NUM — Systèmes numériques
Projet MC2

Rapport de recherche n° 6383 — Décembre 2007 — 16 pages

Abstract: We propose a method to reduce the computational effort to solve a partial differential equation on a given domain. The main idea is to split the domain of interest in two subdomains, and to use different approximation methods in each of the two subdomains. In particular, in one subdomain we discretize the governing equations by a canonical scheme, whereas in the other one we solve a reduced order model of the original problem. Different approaches to couple the low-order model to the usual discretization are presented. The effectiveness of these approaches is tested on numerical examples pertinent to non-linear model problems including the Laplace equation with non-linear boundary conditions and the compressible Euler equations.

Key-words: low-order models, domain decomposition, compressible flows

[*] INRIA Futurs, Equipe-Projet MC2 and Institut de Mathématiques de Bordeaux, UMR 5251 CNRS, Université Bordeaux 1, 33405 Talence cedex, France.

[†] Dipartimento di Ingegneria Aeronautica e Spaziale, Politecnico di Torino. 10129 Torino, Italy

Unité de recherche INRIA Futurs
Parc Club Orsay Université, ZAC des Vignes,

Méthodes Itératives pour la Réduction de Modèles par Décomposition de Domaine

Résumé : On propose une méthode pour réduire les efforts de calcul pour résoudre une équation aux dérivées partielles sur un domaine donné. L'idée principale est de diviser le domaine considéré en deux sous-domaines, et d'employer différentes méthodes d'approximation dans chacun des deux sous-domaines. En particulier, dans un des sous-domaines l'équation en question est discrétisée par une méthode canonique, tandis que dans l'autre un modèle d'ordre réduit du problème original est utilisé. Des stratégies différentes pour coupler le modèle d'ordre réduit á la discrétisation habituelle sont présentés. L'efficacité de ces approches est testée sur des exemples numériques pertinentes pour des problèmes modèles non linéaires, notamment l'équation de Laplace, avec des conditions limites non linéaires, et les équations d'Euler compressibles.

Mots-clés : modèles réduits, décomposition de domaine, écoulements compressibles

1 Introduction

In this contribution we are concerned with the coupling between a full order simulation and a reduced order model. The idea is to reduce the extent of the domain where we perform a canonical numerical simulation by introducing a low-order model which describes the solution far from the region of interest. By reducing the extent of the domain we aim at reducing the costs in terms of required memory as well as in terms of computational time.

In a broad sense, there exist many applications where far from the boundary the solution is weakly dependent on the details of the boundary geometry. In such regions we use a reduced order model based on proper orthogonal decomposition (POD) [4] to solve the problem. This approach allows a representation of the solution by a small number of unknowns that are the coefficients of an appropriate Galerkin expansion. Therefore away from a narrow region close to the boundary of interest the number of unknowns to be solved for is drastically reduced. This idea was previously explored in the context of transonic flows with shocks [3], [2]. Here we extend those works by adapting to that context some classical domain decomposition techniques.

In the following we discuss three possible methods to do that. The first is based on a Schur iteration where the solution of the low-order model is obtained by a projection step in the space spanned by the POD modes. The second is in the same spirit but instead of a Dirichlet-Neumann iteration we employ a Dirchlet-Dirchlet iteration in the frame of a classical Schwarz method. The last approach is of different nature since the solution of the low-order model is not simply based on a projection in the space of the POD modes. It takes into account in a weak sense the governing equations by minimizing the residual norm of the canonical approximation in the space spanned by the POD modes.

The numerical demonstrations shown in the following are relative to two models: the Laplace equation with non-linear boundary conditions modeling radiative heat transfer and the compressible Euler equations in a nozzle. Since this method can be of interest for optimal design applications, where many different geometries must be tested to improve performance, in some cases we have explored the idea of simulating by usual discretization methods just the region where the geometry changes, modeling the rest by POD.

Like all other approaches based on POD, a solution database is necessary to build the basis functions, therefore this method will be useful when many computations for relatively similar cases are to be performed, like for example in shape optimization, see [1].

2 Solution by projection of the solution trace in the space spanned by the POD modes

2.1 Approximation of the Steklov-Poincaré operator by POD

In order to explain the method we take a particular case. Let us consider the Laplace equation $\Delta u = 0$ defined inside a square $\Omega = [0,1] \times [0,1]$. Let $d \in [0,1]$, $\Omega_1 = [0,d] \times [0,1]$, $\Omega_2 = [d,1] \times [0,1]$ and $\Gamma = \overline{\Omega}_1 \cap \overline{\Omega}_2$ the interface between the two sub-domains, see fig. 1. We have Dirichlet conditions on the right boundary (u_R) as well as on the upper (u_U) and lower (u_D) boundaries.

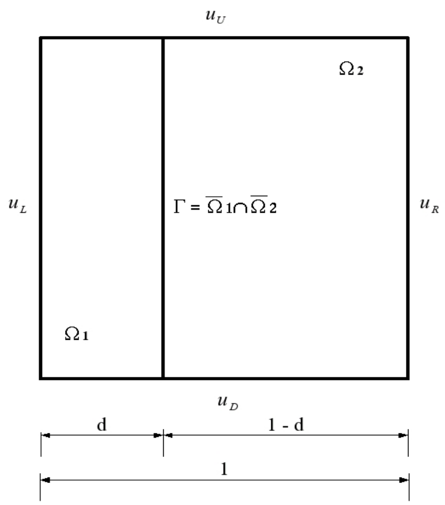


Figure 1: *Problem set-up: Schur*

We want to solve this problem for different values of the Dirichlet data on the left (u_L) boundary. To that end we build an appropriate solution database for a given set of boundary conditions on the left side. In particular, the Dirichlet data on the left boundary is denoted by g^k, $1 \leq k \leq N$.

Let the functions $u_\Omega^{(k)}$ be the discrete solutions of the boundary value problem posed in Ω for different k, and let $\bar{u}_2$ be an harmonic function, restricted to Ω_2, defined as follows: $\bar{u}_2 = 1/N \sum_{k=1}^N u_\Omega^{(k)}$. We compute a Galerkin base of the form $\phi_i = \sum_{k=1}^N b_{ik}(u_2^{(k)} - \bar{u}_2)$ where $u_2^{(k)}$ is the restriction of $u_\Omega^{(k)}$ to Ω_2. The coefficients b_{ik} are found by POD as explained in [8]. It can be shown that this base gives by construction an optimal representation of the original data set $u_\Omega^{(k)}$.

Let us define $\hat{u}_2 = \bar{u}_2 + \sum_{i=1}^M a_i \phi_i$, where M is much smaller than the number of discretization points in Ω_2. For an arbitrary Dirichlet condition on the left boundary of Ω, we want to determine the discrete solution by a canonical approximation in Ω_1 and by the above defined Galerkin representation in Ω_2.

One simple way to implement this idea is to solve the problem by Dirichlet-Neumann iterations. To this end, we follow the steps below:

1. solve the problem in Ω_1 by any discretization method (FD, FEM, etc.), imposing Neumann b.c. on Γ;

2. on interface Γ project the trace of the above solution in the subspace spanned by the traces of the POD modes ϕ_i;

3. recover $\hat{u}_2$ as the prolongation of the trace of $\hat{u}_2$ on Γ inside Ω_2 by using the POD modes;

4. set $\partial \hat{u}_1/\partial n = \partial \hat{u}_2/\partial n$ on Γ;
5. goto (1) until convergence is attained.

This is just one possible solution algorithm, corresponding to a classical domain decomposition method (Schur complement). Another approach consists in solving the problem all at once, as detailed in the following. Let us define A_1 the discretized operator acting on u_1, the restriction of the unknowns belonging to Ω_1; A_Γ the discretized operator acting on u_Γ, the unknowns belonging to Γ and A_2 the discretized operator acting on u_2, the restriction of the unknowns belonging to Ω_2.

The discretized non-linear problem in Ω can be written

$$\begin{pmatrix} A_1 & B_1 & 0 \\ B_1^t & A_\Gamma & B_2^t \\ 0 & B_2 & A_2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_\Gamma \\ u_2 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_\Gamma \\ f_2 \end{pmatrix} \tag{1}$$

where B_1 and B_2 are appropriate interface matrices and f_1, f_Γ, f_2 take into account the boundary conditions.

From (1) we have

$$\begin{aligned} A_1 u_1 + B_1 u_\Gamma &= f_1 \\ B_1^t u_1 + (A_\Gamma - B_2^t A_2^{-1} B_2) u_\Gamma &= f_\Gamma - B_2^t A_2^{-1} f_2 \end{aligned} \tag{2}$$

The matrix $A_\Gamma - B_2^t A_2^{-1} B_2$ is the discrete counterpart of the Steklov-Poincaré operator for Ω_2, see [6]. Consider now the second step of the solution algorithm proposed above. Let $a \in \mathbb{R}^M$ be a vector of components $a_1 \dots a_M$ and $c \in \mathbb{R}^M$ a vector of components $c_1 \dots c_M$. Posing φ_i the trace of ϕ_i on Γ, we take

$$a = \arg\min_{c \in \mathbb{R}^M} \left(\left|\left| u_\Gamma - \sum_{k=1}^{M} c_k \varphi_k \right|\right| \right) \tag{3}$$

where $||\cdot||$ is the norm induced by the canonical l^2 scalar product, noted by $(\cdot, \cdot)$.

Solution of (3) reduces to the solution of the linear problem $\sum_{i=1}^{M} a_i \left(\varphi_i, \varphi_j\right) = (u_\Gamma, \varphi_j)$, $1 \leq j \leq M$. Therefore $a_i = (u_\Gamma, P_i)$, where $P_i = \sum_{j=1}^{M} \left[(\varphi_i, \varphi_j)\right]^{-1} \varphi_j$ is a constant vector computed once for all from the POD modes.

At this point we approximate u_2 with $\hat{u}_2$ and substitute in (1). Since $B_2^t \hat{u}_2 = B_2^t \bar{u}_2 + \sum_{i=1}^{M} a_i B_2^t \phi_i$, we have $B_2^t \hat{u}_2 = B_2^t \bar{u}_2 + \sum_{i=1}^{M} B_2^t \phi_i (u_\Gamma, P_i)$. Finally, letting $\hat{S}_2 = \sum_{i=1}^{M} B_2^t \phi_i P_i$ we obtain the approximation of (2)

$$B_1^t u_1 + (A_\Gamma - \hat{S}_2) u_\Gamma = f_\Gamma - B_2^t \bar{u}_2 \tag{4}$$

where $B_2^t \bar{u}_2 \equiv B_2^t A_2^{-1} f_2$. Matrix $\hat{S}_2$ is the approximation of the discrete Steklov-Poincaré operator obtained by the POD expansion. Equations (2) can of course be solved simultaneously by a standard linear solver.

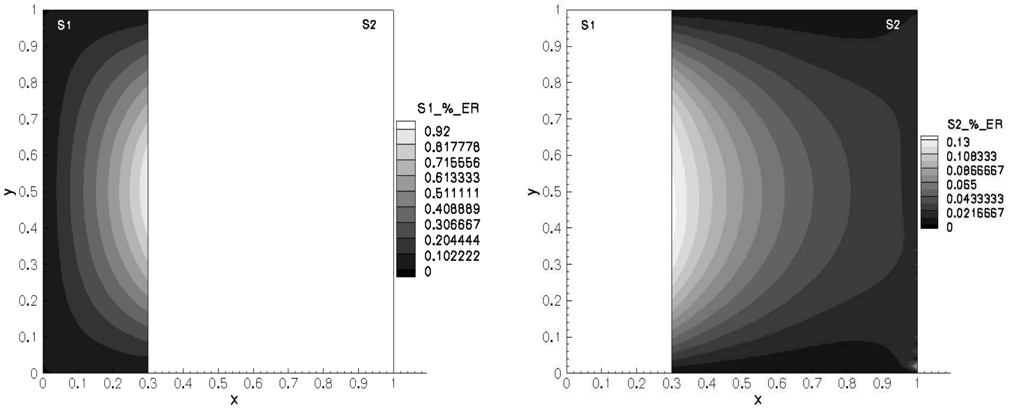
Just like for the usual Steklov-Poincaré operator, (4) amounts to a non-local boundary condition for the problem posed in Ω_1. The main advantage of this approach compared to computing explicitly S_2 is that we do not need A_2^{-1} to build $\hat{S}_2$.

In the following we present some numerical applications. The first case is a plain application of the Schur complement as it was explained above to a non-linear case. The second case is based on the Schwarz method. The third numerical experience is a variant of the all at one method.

2.1.1 Schur complement

A second order finite differences (FD) method coupled to a fix point iteration is used to solve the Laplace equation inside the square domain shown in fig. 1. The left Dirichlet boundary condition is varied to build the needed database. In particular, $u_L = \sin(k\pi y) + y$, $1 \leq k \leq 49$. The boundary conditions on the other sides are, referred to fig. 1, the following: on $u_U : u = 1$, on $u_D : u = 0$ and on $u_R : u^4 - u_0^4 + \frac{\partial u}{\partial n} = 0$.

The domain is split at $d = 1/3$. Then, the POD basis functions are generated on Ω_2 using the previously computed database. In order to check the accuracy of the method, a boundary condition which was not included in the database used to build the POD modes is imposed on the left boundary: $u_L = y^2$, and a second order FD method is used to solve the problem in Ω_1. We use 6 POD modes to recover $\hat{u}_2$ inside Ω_2. Figure 2 presents the result of the test by means of the distribution of the relative error between the FD solution on the entire domain and the approximate Schur complement approach.



(a) (b)

Figure 2: *Distribution of the relative error between the solution obtained by the present method and the solution obtained by a second order FD method on the whole domain.* Ω_1 *(a),* Ω_2 *(b).*

2.1.2 Schwarz method

In the following, a convergent-divergent domain Ω is considered. As before, Ω is divided in two subdomains, Ω_1 and Ω_2, in a way that there exist an overlap region $\Omega_{ov} = \overline{\Omega}_1 \cap \overline{\Omega}_2$ shared by both subdomains (see fig. 3). The Laplace equation is solved with the boundary conditions detailed below to generate a database of k solutions with $1 \leq k \leq 60$, this time varying the geometry of Ω_1. The solution is obtained by means of the finite element method as implemented in [5], using P1 elements on a triangular non-structured mesh and a fixed point iteration. The boundary conditions imposed are: on $u_L : u = \frac{1}{3} \cdot y$, on $u_U : u = 1$, on $u_D : u = 0$ and on $u_R : u^4 - u_0^4 + \frac{\partial u}{\partial n} = 0$.

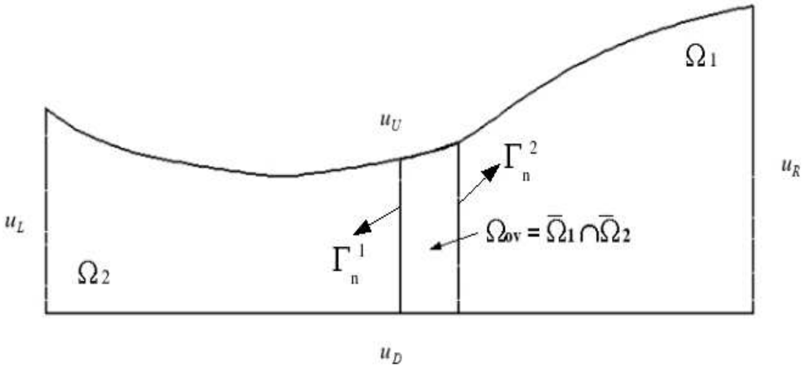
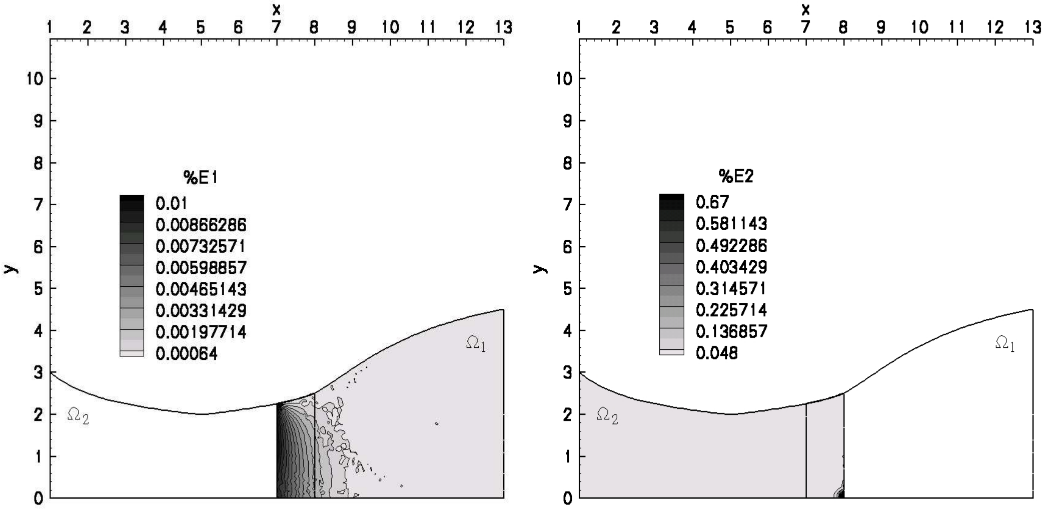


Figure 3: *Problem set-up: Schwarz*

For a geometry of the divergent part which is not included in the database, the solution is determined following a similar approach to that described for the Schur complement but employing this time the classical Schwarz method (see, for example, [6]):

1. solve the problem in Ω_1 by any discretization method imposing Dirichlet b.c. on Γ_n^1;
2. on Γ_n^2 project the trace of the above solution in the subspace spanned by the trace of the POD modes ϕ_i;
3. recover $\hat{u}_2$ as the prolongation of the trace of $\hat{u}_2$ on Γ_n^2 inside Ω_2 by using the POD modes;
4. set $u_1 = \hat{u}_2$ on Γ_n^1;
5. goto (1), $n = n + 1$, until convergence is attained.

Four POD modes are used to recover $\hat{u}_2$ inside Ω_2. Figure 4 shows the results obtained for this case, again in terms of the relative error between a numerical solution (FEM P1) on the entire domain and the approximate Schwarz method.



(a) (b)

Figure 4: *Percentage error distribution on (a) Ω_1 and (b) Ω_2. The reference solution is obtained by FEM P1 elements on a triangular mesh.*

2.1.3 Newton method

In this case the two-dimensional compressible Euler equations are solved

$$\begin{array}{rcl} 0 & = & -\mathbf{v}\cdot\ \mathbf{grad}\, S \\ 0 & = & -\mathbf{v}\cdot\ \mathbf{grad}\, a - \frac{\gamma-1}{2} a\, \mathbf{div}\,\mathbf{v} \\ 0 & = & -\ \mathbf{grad}\,\mathbf{v}\cdot\mathbf{v} - a\left(\frac{2}{\gamma-1}\ \mathbf{grad}\, a - \frac{1}{\gamma(\gamma-1)} a\ \mathbf{grad}\, S\right), \end{array} \tag{5}$$

with $\gamma = 1.4$ for air and being S the entropy, a the speed of sound and $\mathbf{v} = [u, v]$ the velocity, in a convergent-divergent nozzle on a structured mesh. We consider shockless flows.

The λ-scheme [9] is used to solve the equations. Total temperature, total pressure and the flow angle are imposed at the inlet; static pressure at the exit and impermeability at the walls. The complete system is solved by Newton iterations. The resulting linear problems are solved by preconditioned GMRES iterations [7].

Using this code, a database of 90 snapshots is computed. The geometry of the divergent part of the nozzle is changed, Ω_1 in fig. 3. Furthermore, the static pressure at the exist is also varied taking uniformly spaced values in the interval $p = [0.94, 0.99]$ with step 0.01. This corresponds to 15 snapshots for each pressure step. Since the flow is shockless, total temperature as well as total pressure are constant across the nozzle. Therefore, in order to solve in Ω_1, the only unknown boundary condition is the flow angle on Γ^1_n. Hence, a low-order representation of the flow angle v/u for the convergent part, i.e., Ω_2, is constructed retaining 10 POD modes.

We consider a case for which the geometry of the divergent part of the nozzle and the static pressure imposed at the exit do not belong to the set used to build POD database. The two subproblems are coupled by a least squares approximation on the overlapping domain. This is just a variant of the Schwarz method presented in the previous section. The only difference is that the two coupled problems are solved all at once by a Newton method.

The relative errors restricted to Ω_1 are reported in table 1 for some of the flow variables. These errors are computed with respect to the numerical simulation on the entire domain, and are quantified in terms of relative error in L^2 norm, i.e., the L^2 norm of the difference between the solution obtained by the present method and the reference solution divided by the L^2 norm of the reference solution.

Variable	$e(u)\%$	$e(v)\%$	$e(Ma)\%$	$e(v/u)\%$
Error	1.1	2.25	1.14	0.75

Table 1: Relative percentage errors (in L^2 norm) for the flow variables: u (horizontal velocity), v (vertical velocity), Ma (Mach number) and v/u (flow angle).

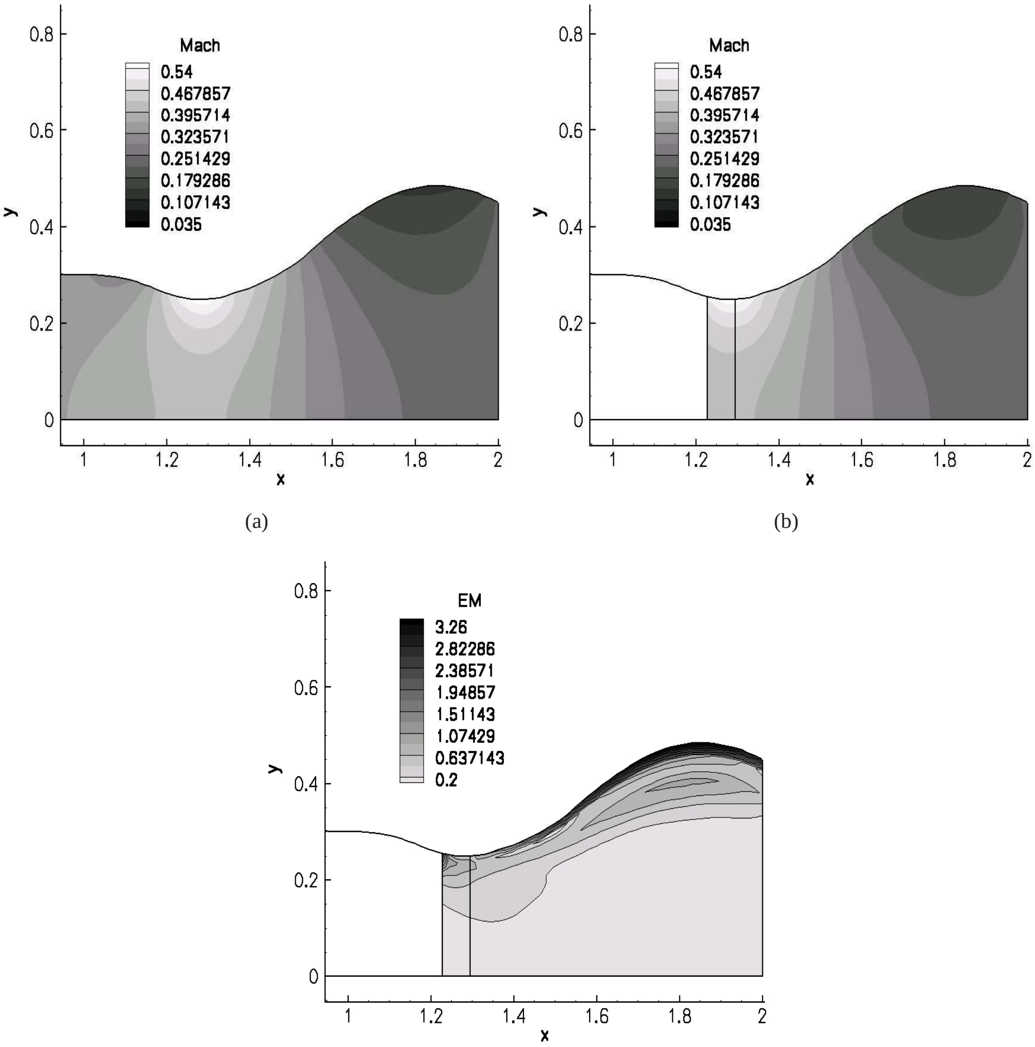
Figure 5(a) presents the distribution of the Mach number obtained by the full numerical simulation on the entire domain. The same quantity obtained when applying the proposed method is shown in fig. 5(b). Finally, fig. 5(c) shows the distribution of the relative percentage error. In figure 6 the flow angle v/u is considered.

3 Solution by minimization of the residual norm in the space spanned by the POD modes

An alternative way to couple the low-order model to a detailed simulation is to look for an approximate solution in the reduced order function space that takes into account the governing equations. Hence, the main difference with respect to the approach described in section 2 is that the approximate solution is found by minimizing the residual norm of a given discretization scheme on the whole domain Ω_2 rather than by projecting the trace of the solution in the space spanned by POD modes.

Again the total pressure and the total temperature are constant across the nozzle and therefore we can write any other variable as a function of the local Mach number and the ratio v/u. In particular let U be the array of the couples $(Ma, v/u)$ for all the grid points belonging to Ω_2. We start by representing this vector in the original Q-dimensional discrete space by a linear combination of basis functions, $U = \sum_{i=1}^{M} \alpha_i \Phi_i$, with $M \ll Q$. The arrays Φ_i have the same structure of U and they have been obtained by POD. The idea is to satisfy the compressible Euler equations (5) in a least squares sense over Ω_2. In other words, let $E(\alpha)$ be the discrete residual of the governing equations as a function of $\alpha = (\alpha_1, \dots, \alpha_M)$ and let

$$I(\alpha) = \frac{1}{2} E^T(\alpha) E(\alpha) \tag{6}$$



(a)

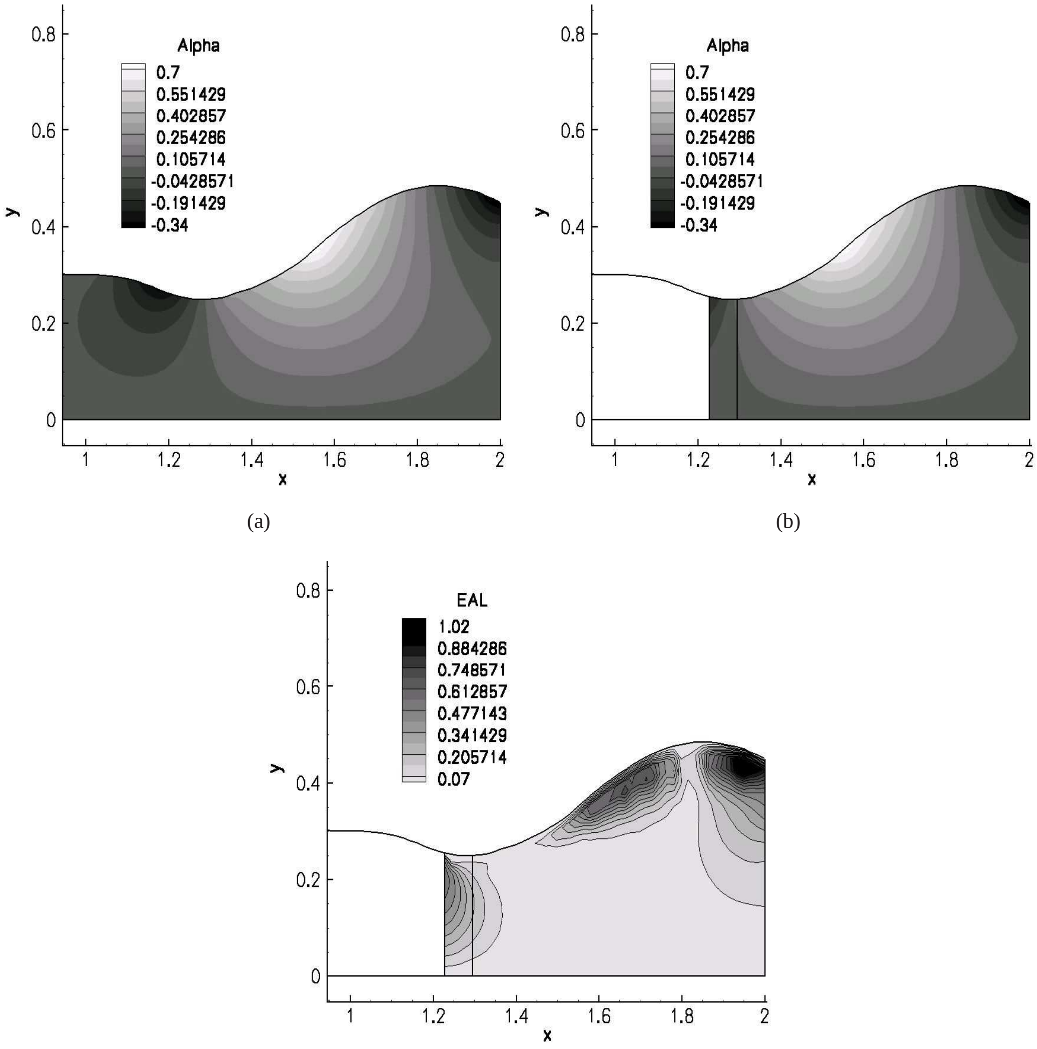
(b)

(c)

Figure 5: *Distribution of $Mach$ number: (a) Numerical solution (λ-scheme). (b) Present method. (c) Relative percentage error distribution on Ω_1.*

be the residual norm. The solution in the POD function space α^* is found by setting

$$\alpha^* = \text{arg min}_\alpha I(\alpha) \tag{7}$$



(a)

(b)

(c)

Figure 6: *Distribution of v/u. (a) Numerical solution (λ-scheme) on the entire domain. (b) Present method. (c) Relative percentage error distribution on Ω_1.*

This is equivalent to a system of non-linear equations

$$\frac{\partial I}{\partial \alpha_i} = \frac{\partial E^T(\alpha^*)}{\partial \alpha_i} E(\alpha^*) = J^T(\alpha^*)E(\alpha^*) = 0 \quad \forall i \in [1, \ldots, M] \tag{8}$$

where J^T is the system's Jacobian.

Equation (8) is solved by a quasi-Newton method where we take

$$E(\alpha^*) \quad = \quad E(\alpha^0) + J(\alpha^0)\Delta\alpha + O(\Delta\alpha^2) \tag{9}$$

and after substituting in (8) we obtain

$$J^T J \Delta\alpha_i = -J^T E \tag{10}$$

Therefore, we are left with the solution of equation (10) at each step of the quasi-Newton algorithm employed to solve (8).

Using the same database generated in Section 2.1.3, two low-order basis restricted to the convergent part of the nozzle are constructed. One for the flow angle v/u and the other for the Mach number. Only 5 POD modes are retained for both the flow angle and the Mach number.

This method is tested for the same case as for the Newton method presented in Section 2.1.3. The only difference is that the solution in Ω_2 is found by minimization of the residuals norm in the space spanned by the POD modes. Otherwise, the full numerical simulation is employed on Ω_1, and a classical Schwarz overlapping method is used to iterate the solution to convergence.

The reference Mach number on the entire domain, the Mach number obtained by the present method on Ω_1 as well as the distribution of the relative percentage error on Ω_1 are shown in figures 8(a), 8(b) and 8(c), respectively.

In fig. 7, the results for the flow angle v/u are shown while table 2 presents the relative errors obtained in Ω_1.

Variable	$e(u)\%$	$e(v)\%$	$e(Ma)\%$	$e(v/u)\%$
Error	0.05	0.26	0.05	0.13

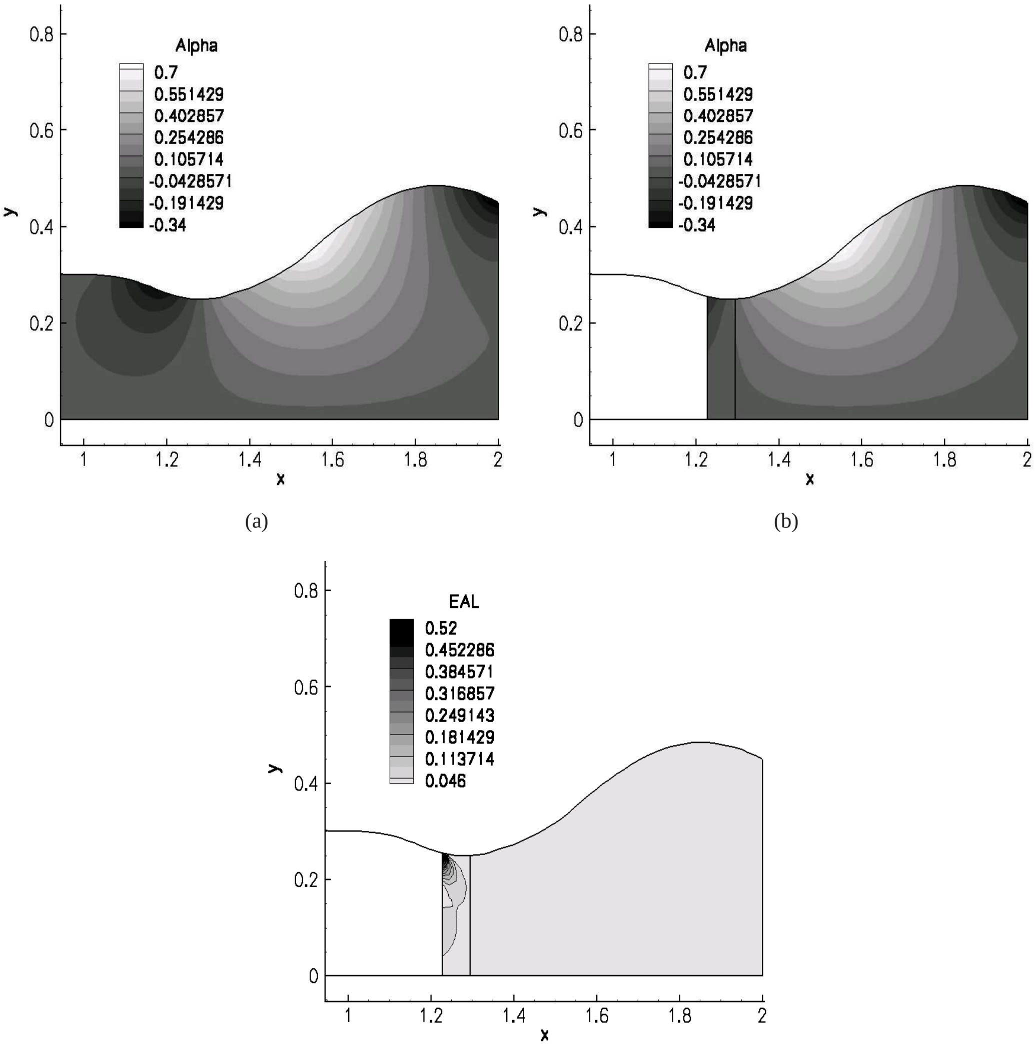
Table 2: Relative percentage errors (in L^2 norm) for the flow variables: u (horizontal velocity), v (vertical velocity), Ma (Mach number) and v/u (flow angle).

These results show a much better accuracy as compared to those of section 2.1.3 as one can conclude by comparing tables 1 and 2.

Finally, in order to asses sensitivity of the results with respect to the minimization problem, we show in table 3 the error in Ω_2 when we use for α the initial guess. In table 4 we present the error at the end of the Schwarz iteration. It is seen that the fact of approximating the solution of the Euler equations with appropriate boundary conditions, even in a crude way, in Ω_2, significantly improves the solution with respect to the initial guess.

Variable	$e(u)\%$	$e(v)\%$	$e(Ma)\%$	$e(v/u)\%$
Error	35.52	45.65	39.28	6.25

Table 3: Relative percentage errors (in L^2 norm) for the flow variables: u (horizontal velocity), v (vertical velocity), Ma (Mach number) and v/u (flow angle).

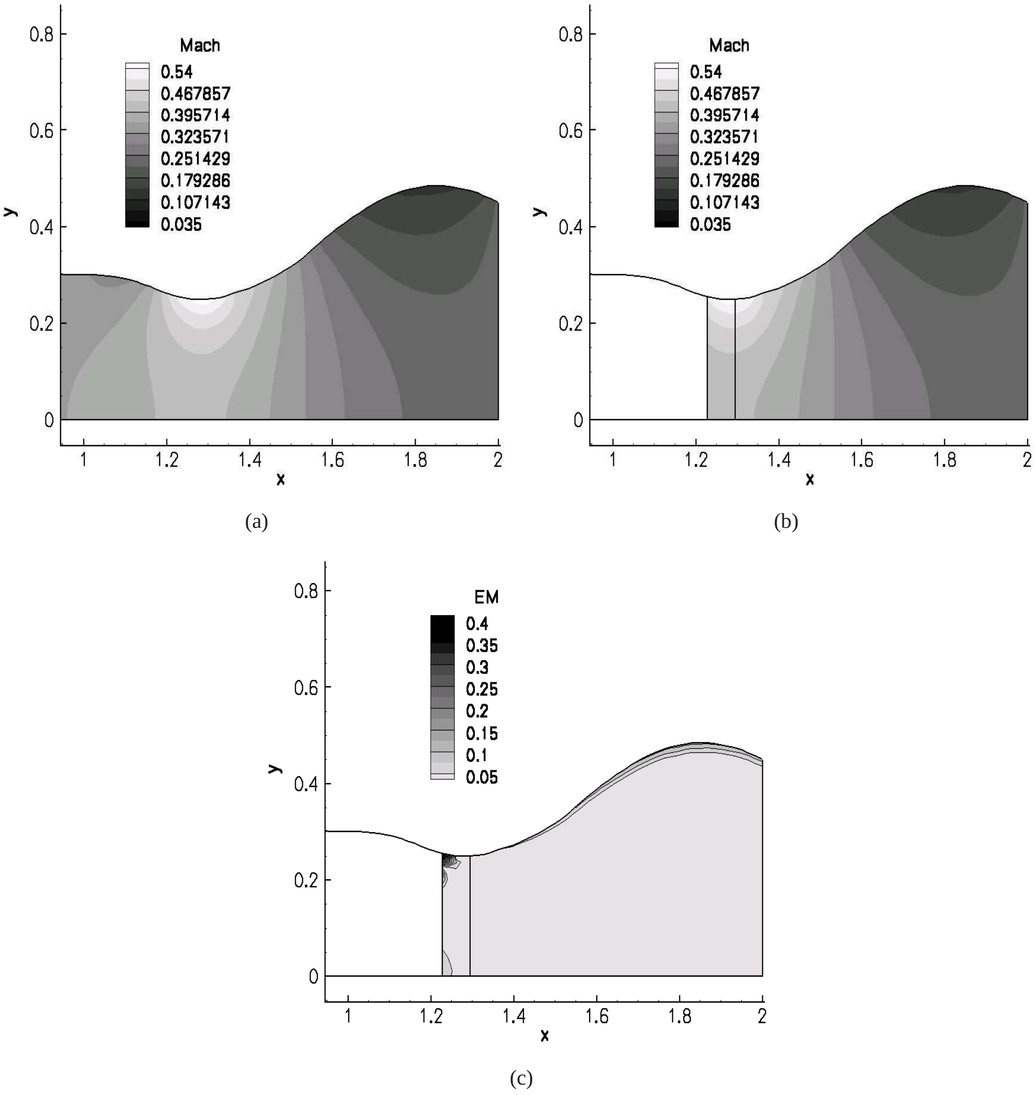


(a)

(b)

(c)

Figure 7: *Distribution of v/u. (a) Numerical solution (λ-scheme) on the entire domain. (b) Present method. (c) Relative percentage error distribution on Ω_1.*



(a)

(b)

(c)

Figure 8: *Distribution of $Macxh$ number on Ω_1. (a) Numerical solution (λ-scheme). (b) Present method. (c) Relative percentage error distribution on Ω_1.*

4 Discussion

The fact of reducing the extent of the computational domain does not guarantee that one can get a solution faster as compared to solving the problem on the entire domain. In particular, let us consider

Variable	$e(u)\%$	$e(v)\%$	$e(Ma)\%$	$e(v/u)\%$
Error	2.01	3.94	2.08	2.90

Table 4: Relative percentage errors (in L^2 norm) for the flow variables: u (horizontal velocity), v (vertical velocity), Ma (Mach number) and v/u (flow angle).

the method described in section 2.1.3. It would in principle be the less demanding in terms of computational time since it just amounts to a non-local boundary condition in the frame of a Newton method. However, this is not necessarily the case since the convergence of the implicit iteration is spoiled by such non-local boundary condition and the number of Newton steps to attain convergence goes from 7 on the whole domain, to 11 when we only solve on Ω_1. The slower convergence rate may lead to comparable costs in terms of CPU time, as a function of the fraction of the domain that is actually resolved. On the other hand, the fact of reducing the number of grid points is reflected almost proportionally on the memory requirements. However, the Jacobian matrix will have a non-sparse block corresponding to the non-local boundary condition induced by the approximation of the Steklov-Poincaré operator.

Concerning the method described in section 3 the cost of each step of the proposed minimization algorithm can be split up as follows. i) The computation of J. The Jacobian is evaluated by one-sided finite differences. Consequently its cost in terms of CPU time and memory requirements is proportional to the dimension of the low-order space M and the cost of computing the residual vector, i.e., cost$(J) = M \times$ cost(E). Since M is $O(10)$, the Jacobian matrix can be formed after few residual evaluations that are cheap in terms of CPU time. Memory requirements could become prohibitive for very big problems only on serial architectures. ii) The computation of the symmetric matrix $A = J^T J$. The dimensions of A are $M \times M$, with $M = O(10)$. The required floating point operations are $(M \times M)/2 \times N$ and they are less than the operations needed for the computation of E. iii) The cost of finding the solution to the linear system of size $M \times M$ can be neglected. Summarizing, it can be deduced that the cost is equivalent to some residual evaluations. The number of iterations needed in order to find the minimum are generally less than five, yielding an overall cost negligible with respect to canonical CFD calculation on Ω_2.

We remark two major limits of these approaches. The first is of course that the results depend to a large extent on the database used for the POD modes. If the configuration under consideration lies in a region of the parameter space far from that explored when building the database, then the approximation error can be large. It is true, however, that the low-order model should be used where the solution does not strongly depend on the boundary conditions or the geometry. Another limitation is that an efficient way to improve the approximation quality comparable for example to grid refinement is not available. In principle, we would like to increase the approximation accuracy by enriching the functional space in which the solution is sought, based on some objective criteria. Unfortunately a general framework for such improvement is not presently available and it is the object of present research.

In conclusion we presented some possible implementations of a method to reduce the extent of the computational domain in the numerical solution of partial differential equations. The idea of using models that take into account different physical phenomena in different subdomains is old. Here we revisited this approach using a low-order model in the framework of classical domain decomposition techniques. The results in terms of the approximation error are promising for all of the cases that we showed. A major challenge to be pursued is to find a viable a posteriori error estimation technique for iteratively adapting the POD function space.

References

[1] T. Bui-Thanh, M. Damodaran, and K.E. Willcox. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA Journal*, 42(5):1505–1516, 2004.

[2] P. A. LeGresley and J.J. Alonso. Dynamic domain decomposition and error correction for reduced order models. In *41st AIAA Aerospace Sciences Meeting & Exhibit, AIAA Paper 2003-0250*, pages 1–13, 2003.

[3] D.J. Lucia, P.I. King, and P.S. Beran. Reduced order modeling of a two-dimensional flow with moving shocks. *Computers and Fluids*, 32:917–938, 2003.

[4] J. L. Lumley. The structure of inhomogeneous turbulent flows. In *Atmospheric Turbulence and Radio Wave Propagation*, edited by A. M. Yaglom and V. L. Tatarski, Moscow, pages 166–178, 1967.

[5] Olivier Pironneau, Frédéric Hecht, and Antoine Le Hyaric. Freefem++. `http://www.freefem.org`.

[6] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Dierential Equations*. Oxford Science Publications, 1999.

[7] Youcef Saad. Sparsekit: a basic tool kit for sparse matrix computations. `http://www-users.cs.umn.edu/∼saad/software/SPARSKIT/sparskit.html`.

[8] L. Sirovich. Turbulence and the dynamics of coherent structures. Parts I,II and III. *Quarterly of Applied Mathematics*, XLV:561–590, 1987.

[9] L. Zannetti and B. Favini. About the numerical modelling of multidimensional unsteady compressible flow. *Computers and Fluids*, 17:289–299, 1989.

Unité de recherche INRIA Futurs
Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399